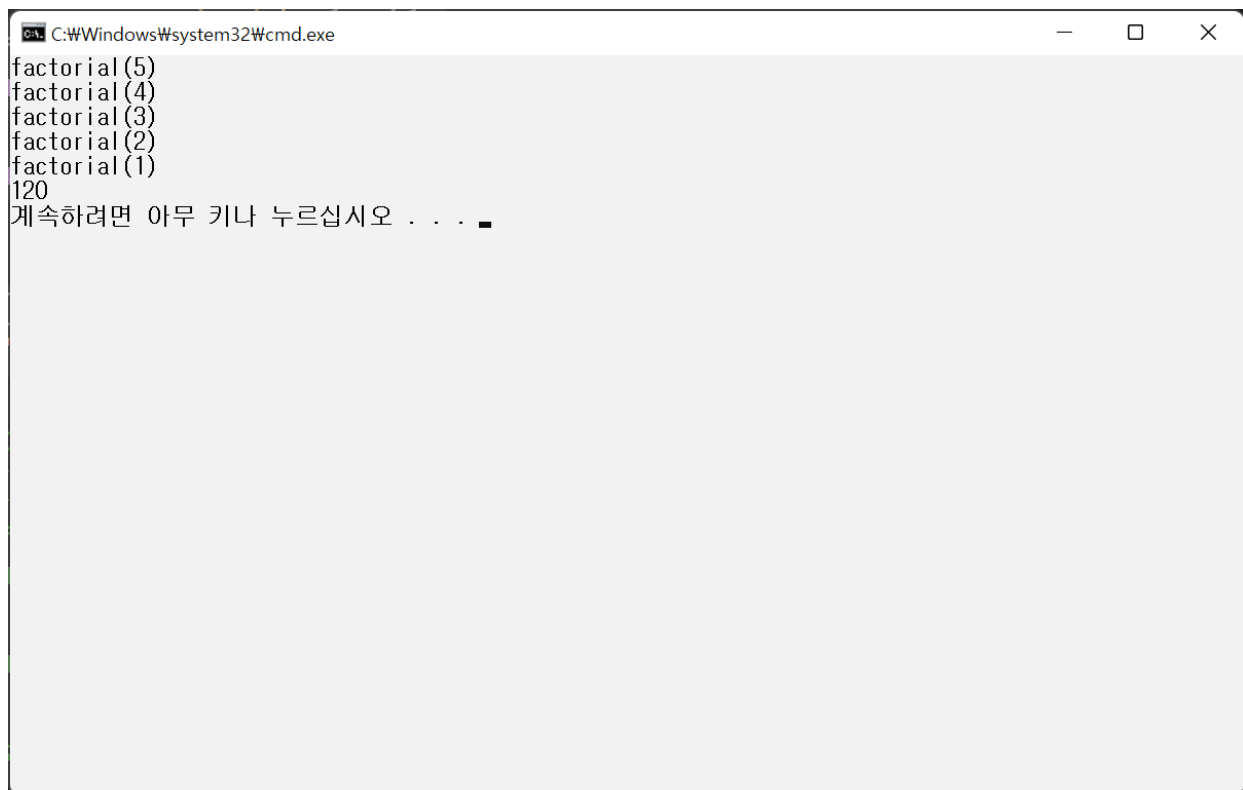


스터디 첫주자는 c언어로 쉽게 풀어쓴 자료구조 2장 순환 파트부터 시작하였습니다. 교재에 있는 문제를 연습문제를 풀어보고 남은 시간 동안은 백준에서 재귀함수와 관련된 문제를 풀어보았습니다.

출력문이 추가된 순환적인 팩토리얼 계산 프로그램

```
#include <stdio.h>
int factorial(int n) {
    printf("factorial(%d)\n", n);
    if (n <= 1)
        return 1;
    else
        return n * factorial(n - 1);
}
int main() {
    printf("%d\n", factorial(5));
}
```



A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window displays the output of the factorial program. The output shows the recursive calls: factorial(5), factorial(4), factorial(3), factorial(2), factorial(1), followed by the final result 120. Below the output, there is a line of Korean text: "계속하려면 아무 키나 누르십시오 . . .".

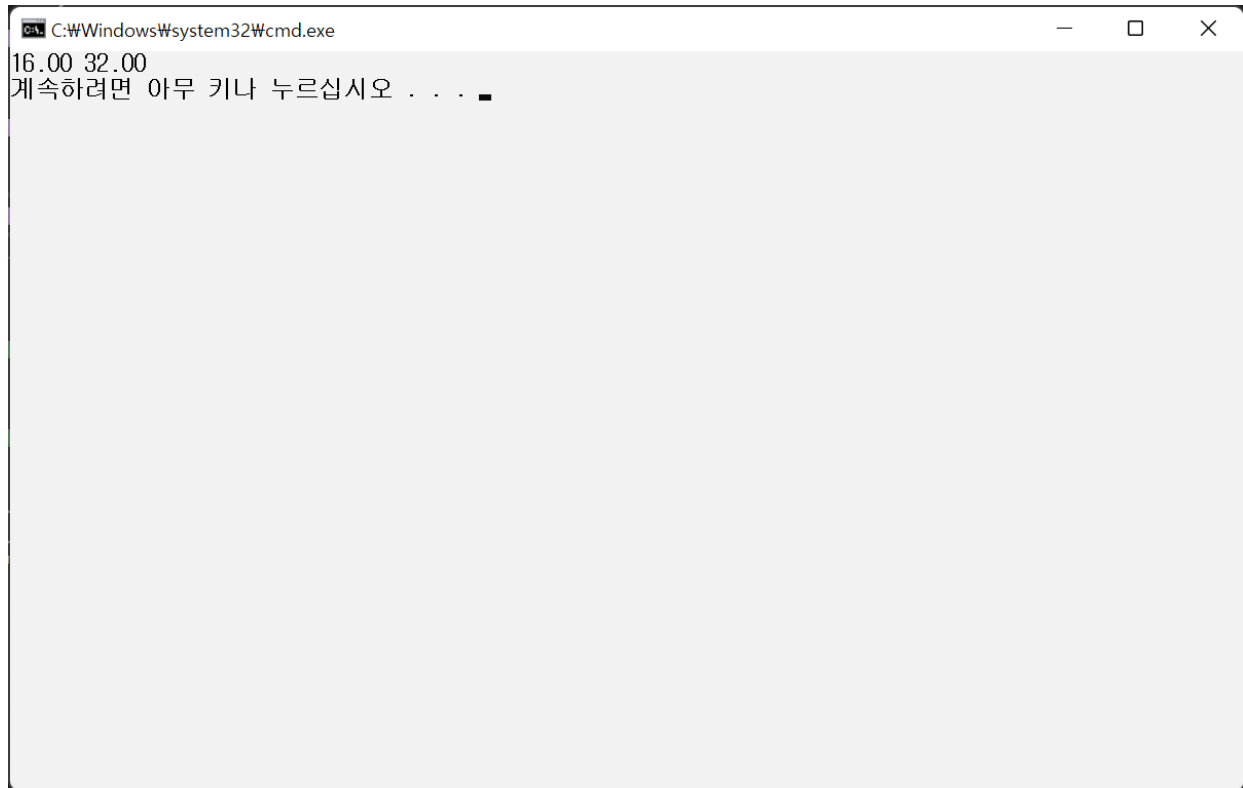
순환적인 거듭제곱 계산 프로그램

```
#include <stdio.h>
double power(double x, int n) {
    if (n == 0)
        return 1;
    else if (n % 2 == 0)
        return power(x * x, n / 2);
    else
        return x * power(x * x, n / 2);
}
//n이 짝수, 홀수의 여부와 상관 없이 거듭제곱값을 계산할 때
```

```

    /*if (n == 0)
        return 1;
    else
        return x * power(x, n - 1);*/
}
int main() {
    printf("%.21f %.21f\n", power(2, 4), power(2, 5)); //2의 4제곱, 2의 5제곱 출력
}

```



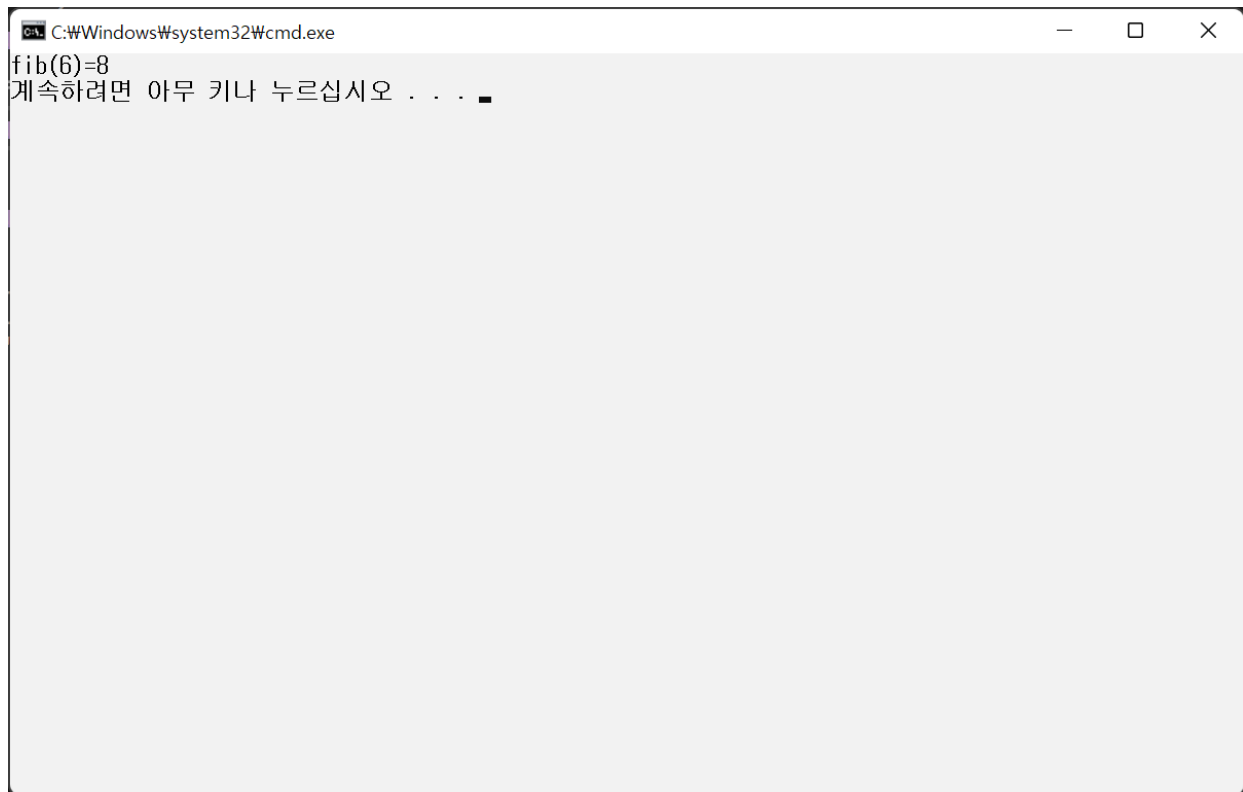
A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window shows the output of the program: "16.00 32.00" followed by a Korean prompt "계속하려면 아무 키나 누르십시오 . . .".

순환적인 피보나치 수열 계산 프로그램

```

#include <stdio.h>
int fib(int n) {
    if (n == 0)
        return 0;
    if (n == 1)
        return 1;
    else
        return fib(n - 2) + fib(n - 1);
}
int main() {
    printf("fib(%d)=%d\n", 6, fib(6));
}

```



피보나치 수열 계산 프로그램을 재귀함수가 아닌 반복적인 방법으로 구현

```
#include <stdio.h>
int fib(int n) {
    int fib = 0, fib1 = 1, fib2 = 0;
    for (int i = 2; i <= n; i++) {
        fib2 = fib + fib1;
        fib = fib1;
        fib1 = fib2;
    }
    return fib2;
}
int main() {
    printf("fib(%d)=%d\n", 6, fib(6));

    return 0;
}
```

```
C:\Windows\system32\cmd.exe
fib(6)=8
계속하려면 아무 키나 누르십시오 . . .
```

하노이탑

하노이탑 문제는 헛갈려서 관련된 문제를 더 풀어봐야 될 것 같습니다.

```
#include <stdio.h>
void hanoi_tower(int n, char from, char tmp, char to) {
    if (n == 1)
        printf("%c에 있는 %d의 원판을 %c로 옮긴다.\n", from, n, to);
    else {
        hanoi_tower(n - 1, from, to, tmp);
        printf("%c에 있는 %d의 원판을 %c로 옮긴다.\n", from, n, to);
        hanoi_tower(n - 1, tmp, from, to);
    }
}
int main() {
    hanoi_tower(4, 'A', 'B', 'C');
}
```

```
C:\Windows\system32\cmd.exe
A에 있든 1의 원판 B로 옮긴다.
A에 있든 2의 원판 C로 옮긴다.
B에 있든 1의 원판 C로 옮긴다.
A에 있든 3의 원판 B로 옮긴다.
C에 있든 1의 원판 A로 옮긴다.
C에 있든 2의 원판 B로 옮긴다.
A에 있든 1의 원판 B로 옮긴다.
A에 있든 4의 원판 C로 옮긴다.
B에 있든 1의 원판 C로 옮긴다.
B에 있든 2의 원판 A로 옮긴다.
C에 있든 1의 원판 A로 옮긴다.
B에 있든 3의 원판 C로 옮긴다.
A에 있든 1의 원판 B로 옮긴다.
A에 있든 2의 원판 C로 옮긴다.
B에 있든 1의 원판 C로 옮긴다.
계속하려면 아무 키나 누르십시오 . . .
```

백준 25501번 재귀의 귀재

팰린드롬: 문자열이 앞에서부터 읽었을 때와 뒤에서부터 읽었을 때가 같은 문자열

isPalindrome 함수는 주어진 문자열이 팰린드롬이면 1, 팰린드롬이 아니면 0을 반환하는 함수

테스트케이스 T를 입력하고 문자열을 입력받아 팰린드롬의 결과와 recursion 함수의 호출 횟수를 출력해야 한다. recursion 함수의 호출 횟수를 반환하는 count()라는 함수를 만들어서 코드를 구현해보려고 했지만 아직 문제를 해결하지 못하였습니다.

```
#include <stdio.h>
#include <string.h>

int recursion(const char *s, int l, int r){
    if(l >= r) return 1;
    else if(s[l] != s[r]) return 0;
    else return recursion(s, l+1, r-1);
}

int isPalindrome(const char *s){
    return recursion(s, 0, strlen(s)-1);
}

int main(){
    // printf("ABBA: %d\n", isPalindrome("ABBA")); // 1
    // printf("ABC: %d\n", isPalindrome("ABC")); // 0
    char s[100];
    int T;
    scanf("%d", &T);
```

```
for (int i = 0; i < T; i++) {  
    scanf("%s", s);  
    printf("%d %d\n", isPalindrome(s), count(s));  
}  
}
```